BIOSYSTEMS ENGINEERING 114 (2013) 306–317

Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110

Research Paper

Applicability of the ISO 11783 network in a distributed combined guidance system for agricultural machines

*Juha Backman**, Timo Oksanen, Arto Visala*

Aalto University, School of Electrical Engineering, Department of Automation and Systems Technology, PO Box 15500, 00076 Aalto, Finland

ARTICLE INFO

Article history:
Received 27 April 2012
Received in revised form
4 December 2012
Accepted 14 December 2012
Published online 9 February 2013

Tractors have developed considerably over recent years. More and more electronics have been added, and nowadays the devices are all able to communicate with one another through a common standardised interface: the *ISO 11783 network*. The standard defines the roles of devices on the network and provides standard interfaces for the devices to obtain access to the services on other devices. The ISO 11783 standard also provides functions for operating the tractor via remote control.

This article discusses the case of a tractor connected to a trailer-type implement, both of which were automated for navigation purposes. The article discusses the requirements for communication architecture to command both the tractor and the implement for guidance. The underlying idea is to handle the guidance system over an ISO 11783 network, which, on the one hand, provides a communication channel, but, on the other hand, also sets limits for information flows. The use of ISO 11783 network in the combined navigation system of a tractor and implement has not previously been reported.

The functionality of the proposed navigation system has been tested and proven to work during different test drives. It was found that it is not possible to distribute the controller due to the requirements of the multivariate control problem and the limits of the ISO 11783 network. However, it is possible to transfer all the measurements and controls through the ISO 11783 network. As a result, the article proposes additional messages for the ISO 11783 standard.

1. Introduction

Commercial solutions for automatic guidance of agricultural machines have been on the market since the early 2000s. Nowadays, automatic guidance systems for tractors are nothing new for farmers. However, many solutions focus only on tractor navigation and omit the implement (Farm Journal Inc., 2011). Even if some solutions support handling hitch-mounted implements with offsets, the common form of agricultural machine, a tractor + trailed implement, is not fully supported. The reason is that a guidance system for a tractor–trailer requires more complex mathematical algorithms and more sensors. In this article, the guidance system that takes the implement into account is referred to as a *combined guidance* system.

A combined guidance system for a tractor and a trailer-type implement typically requires at least two positioning sensors. Both of them may be based on Global Navigation

* *Corresponding author*. Tel.: +358 9 470 25154; fax: +358 9 470 23308.
E-mail address: juha.backman@aalto.fi (J. Backman).

1537-5110/$ – see front matter
http://dx.doi.org/10.1016/j.biosystemseng.2012.12.017

Satellite System (GNSS), so that the position of each is known in the global coordinate system; or alternatively, can measure the position relative to the previous swath or to some other marks in the field. An *active implement guidance* system also requires more than just steering the tractor wheels (front, rear, articulated); it also requires some controllable degree of freedom on the implement. The actuation may be in the form of the side shift of an implement, an articulated drawbar or steering wheels on an implement. A *passive implement guidance* system has no controllable degree of freedom on the implement, but the position is still controlled through the tractor's steering capabilities.

There are many manufacturers of agricultural machinery, several for agricultural tractors and even more for the implements. The number of combinations is countless. In order to share information, a standard ISO 11783 'Tractors and machinery for agriculture and forestry – Serial control and communications data network' has been prepared, and it is nowadays widely accepted within the industry. The standard series currently contains 13 standards, parts 1–13, while one more is currently under preparation.

A necessary requirement for a combined guidance system is the interoperability of machines manufactured by different companies. The ISO 11783 standardises communication between the tractor and the implement, but unfortunately the current standard does not support a combined guidance system in a general sense. The other requirement, which is more practical, is that three brands (tractor A, implement B, guidance system C) will be supported in the architecture.

The objective of this article is to discuss and present the ways in which a combined guidance system can be realised by using the ISO 11783 network, and to discuss how the ISO 11783 network should be improved in order to support a combined guidance system. The more specific objective is to propose a *decentralised* and *generic* guidance system for a tractor–implement system. Within this context, generic guidance system is defined in such a way that the implement's kinematics is not limited, e.g., to a trailer with steering wheels but other kinematics are also considered. Also within this context, the decentralised guidance system is defined in such a way that the measurement instruments and the actuators are not directly wired to the navigation controller and are not necessarily provided by the navigation system supplier. This study does not consider how to decentralise the navigation controller or algorithm through standard interfaces into a multi-vendor, tractor–implement system.

The specific challenges related to the objectives include the compatibility of the measurement and control messages for different kinds of tractor–implement systems, the real-time implementation of the navigation system and the information flow between the navigation components. It is evident that in the case of the decentralised guidance system, more information than just the sensor and control are required, whereas, within the context of the control, some state information needs to be exchanged, too.

In this article, a tractor and trailer-type implement are used for illustration purposes, but the results are not limited to that kind of kinematics. The system has been used to realise a combined control system, but the methodology for the navigation algorithms is not discussed in this article. For more details on algorithms, see Backman, Oksanen, and Visala (2010, 2012a).

This first part of the article reviews the history of the guidance systems and discusses the ISO 11783 standard. The second part derives the requirements for information exchange in a decentralised combined guidance system by using the tractor–implement as an example. The third part discusses the physical, logical and software architecture of a generic combined guidance system. In the fifth part the test results and user experience with the proposed system are presented. Finally, the last part contains discussion and conclusions about the research.

2. Background

2.1. Development of autonomous vehicles

Scholars have been doing research on autonomous vehicles for a long time. There are many famous and long-term projects where such intelligent vehicles have been, or are being, developed. One of the first truly autonomous cars was the *VaMP*. It was able to drive on the basis of machine vision without human intervention for more than a hundred kilometres on the German *Autobahn* (Dickmanns, 2007). The Carnegie Mellon University Navigation Laboratory has built driverless car platforms (*NavLab*) for the development of navigation algorithms (Omead, 1990). Probably the most famous competitions between autonomous vehicles are the Darpa Grand Challenge and the Darpa Urban Challenge (Darpa, 2007). The vehicles that have taken part in those competitions are quite advanced, having many sensors for detecting the road and other elements of the environment, such as obstacles. In addition, many computers are also required to process all that information and to navigate the car.

Autonomous vehicles for agricultural operations have also been under development for almost three decades now. The difference between them and road vehicles is that something can be assumed about the conditions for agricultural operations. The field usually has quite an even surface and moving obstacles are not usually present, except other working machines. The positions of other working machines are also usually known with a high degree of accuracy. Furthermore, the boundaries of the field plots are mapped and the environment is structured in this sense. The main limiting factor is that the manufacturing price has to be much lower than for the cars in the Darpa competitions in order to be reasonable for real-life applications. The accuracy requirements are, however, quite high: usually less than a 10 cm relative error between adjacent driving lines.

There are different approaches to building an autonomous agricultural vehicle. One is to build it completely from scratch. Another approach is to take a commercial product and modify it. A third approach is to use a commercial product as it is and to add navigation as an *accessory* to it. One project in which the agricultural robot was built completely from scratch was the *Modulaire* platform (Rintanen et al., 1996). The Modulaire platform was a tracked off-road vehicle. It had a real-time kinematic Global Positioning System (RTK GPS) and a fibre-optic gyro for navigation purposes. Another similar robot was the *Weedy* (Ruckelshausen et al., 2006), a four-wheel,

steered robot for mechanical weed control. It did not, however, use GPS to measure its position. Instead, it had a colour camera to track plants and a gyroscope for headland turnings.

HortiBot represents the second approach, in which a radio-controlled slope mower was transformed into a tool carrier robot (Jørgensen et al., 2006). Also, Blackmore, Griepentrog, Nielsen, Nørremark, and Resting-Jeppersen (2004) developed an autonomous tractor from a small garden tractor. The irony was that it required two persons to operate it: one to give it instructions and another for safety reasons. Nagasaka (2009) has developed an autonomous rice transplanter.

The third approach was to equip a standard tractor for autonomous operations. Lenain, Thuilot, Cariou, and Martinet (2005, 2006) concentrated more on path tracking and position estimation, but they used a standard tractor as a test platform. Strentz, Dima, Wellington, Herman, and Stager (2002), for their part, focused on semi-autonomous tractors for spraying applications. Commercial products for automatic guidance or automatic steering are also available (John Deere, 2012; AGCO, 2012). These commercial products are intended to follow predefined paths or adjacent driving lines and, therefore, do not include any planning or reactive capabilities.

2.2. Control strategies of the autonomous vehicles

With autonomous vehicles, the software architecture and control strategies have also improved tremendously over the years. The basic and earliest strategy for controlling a robot was *sense-model-plan-act*. First, the robot senses the environment and stores the information into some kind of database. Then, the robot's movements are planned according to this database. Finally, the first step of the plan is executed and the robot moves. The benefit of this approach is that the robot's behaviour is predictable and the researcher can know beforehand what should happen. The drawback is that this may require a great deal of computer power. Situations may change during the planning phase and something catastrophic may occur (Murphy, 2000).

Nowadays, a more popular strategy is *reactive control*. Scientists have studied the behaviour of animals and extracted some primitive laws that animals follow. These laws, or behaviours, are then adapted to fit the robots. The difference between this strategy and the first control strategy is that the planning phase is skipped. The sensing elements launch some actions directly as a result of some sensing input. The benefit of this approach is that it is computationally light and easy to develop. The drawback, however, is that the movement of the robot is not always predictable. Another drawback is that, without any higher-order planning, the robot may end up in situations that it cannot handle (Murphy, 2000).

The solution to these problems is to use both strategies. The control task is divided into layers. The reactive part is used whenever possible, whereas the planning part is launched depending on the situation. This hybrid control strategy exploits the best features of both strategies (Murphy, 2000). In order to make a system that is both reactive and capable of planning, some kind of hierarchical architecture is required. The Autonomous Robot Architecture (AuRA) was the first navigation system to use this kind of hybrid architecture (Arkin & Balch, 1997).

For agricultural robots, Blackmore, Have, and Fountas (2002) have proposed an object-oriented architecture with the message passing through a common bus. Tasks are divided into subprograms called *agents*, which can be replaced or modified to fit a certain application. The hardware is abstracted into a Hardware Abstraction Layer (HAL) agent interface between the software and the devices that will be operated on. This agent also includes the critical control loops. Such control loops are, for example, steering and speed controllers. These controllers utilise the inverse kinematic model to calculate the proper control values. This HAL agent can, in a way, be considered the reactive part and all above that belongs to the planning part.

Although the above-mentioned system has the desired structure, the control of the vehicle can be more precise. One way to utilise the kinematic model more effectively is to use Model Predictive Control (MPC) (Maciejowski, 2002). MPC predicts the future according to the model of the system and tries to minimise some given criterion, while also taking into account the model's restrictions. The criterion is usually a sum of the squared errors of the desired and actual output values. Because in this case the model is nonlinear, a Nonlinear Model Predictive Control (NMPC) is used. Backman et al. (2012a) have discussed the control problems pertaining to NMPC in more detail. This is, however, taken into account when designing the software architecture and physical requirements of the navigation equipment.

Another restriction of the above-mentioned proposed system architecture for an autonomous tractor is the requirement for a proprietary bus. The messages in the bus are not compatible with different manufacturers and are not harmonised with the ISO 11783 standard. Ehrl and Auernhammer (2007) have proposed a Steer-by-Wire approach via the ISO 11783 network and discussed the requirements and applicability of the bus. While their results were promising, they also noted that further investigation is needed with, for example, predefined bus load scenarios. Also, they left the message content open and did not investigate the combined tractor–implement navigation system.

3. The ISO 11783 series of standards

The ISO 11783 series of standards, 'Tractors and machinery for agriculture and forestry – Serial control and communications data network', was developed to support the exchange of information between different manufacturers' mobile agricultural machinery products. The need for a communication standard is evident; in a typical agricultural machine configuration, a tractor is connected to one or more implements that are manufactured by a different company from the one that manufactured the tractor.

The ISO 11783 standard is partially based on the *SAE J1939* standard (SAE J1939:1994), which was developed for use in truck and bus applications. Both standards are based on the Bosch Controller Area Network (CAN) 2.0B specification (CAN Specification Version 2.0:1991). The purpose of the ISO 11783 standard is to 'specify the method and format of transfer of data between sensor, actuators, control elements, information storage and display units whether mounted or part of the

tractor, or any implements'. The market name for systems and devices that are proven to be compatible with the standard is *ISOBUS*.

Nowadays, the standard contains 13 parts. Part 1 is a general introduction to the standard series and it includes definitions and abbreviations. Parts 2–5 and 12 specify the lower-level protocol, or protocol stack. The other parts specify the higher-level protocol for various applications in the network. Part 6 specifies *virtual terminal* and the protocol for the corresponding client. Part 7 specifies implement messages, basically those used for tractor–implement communication. Part 8 covers the drivetrain. Part 9 specifies a tractor as a device in the network and a Tractor Electronic Control Unit (TECU). Part 10 specifies *task controller* and the protocol for the corresponding client as well as the data file format for the *tasks*. Part 11 is nowadays an online dictionary for the presentation layer, which is mainly used for part 10. Part 13 specifies *file server* and the protocol for using it.

Guidance-related material can be found in part 7, which contain 'remote control' messages of tractor and some implementation issues pertaining to remote control messages. With the remote control message, an implement may command the *curvature* of the tractor to the desired value; it is up to the tractor's internal control system to operate the tractor's steering hydraulic cylinder to realise the setpoint. The other message gives feedback to an implement about the estimated or measured curvature. Similar messages are also given for the *speed* command and measurement.

However, it is not mandatory that a tractor manufacturer implements the remote control messages. Part 9 specifies tractor classes, from 1 to 3, and only in Class 3 are the remote control messages required, but guidance remote control is still an option (ISO 11783-9:2002).

Even if the remote control for a tractor's curvature is available in a standard format, it will only cover one crucial link in the combined guidance system. The missing links are between the guidance system and the implement, and state information interchange between the tractor and the implement. Hence, in the sense of the combined guidance system, the ISO 11783 is considered more like a limiting factor than a supporting feature. The standard needs to be improved to support a combined guidance system.

4. The test configuration

For better illustration of information flows, the system used to test a combined guidance system is presented. The test configuration consisted of a tractor and a towed, trailer-type implement. The tractor was a *Valtra T132*, equipped with an ISO 11783 compatible Tractor ECU and a Class 3 guidance option. Thus, the ISO 11783:7 remote control messages could be used to steer the tractor and to control the speed. The implement was a *Junkkari Maestro 3000* seed drill equipped with an ISO 11783 compatible implement controller. The seed drill was a trailer-type seed drill; the supporting wheels were located in the rear and the coulters just in front of them. For the guidance system, the drawbar of the seed drill was modified by adding an articulated joint to the end of the seed drill. The articulated joint gave an additional degree of freedom for guiding the vehicle. Hence, the combined guidance system had three actuators under control: the steering angle of the front wheels of the tractor, the angle of the articulated drawbar of the seed drill and the speed of the tractor. The test configuration, i.e. the vehicle, is presented in Fig. 1.

The objective of the guidance system in agricultural operations, including sowing, is to control the trajectory of the vehicle, to keep it within a constant distance of the adjacent driving line. Or, in agricultural terms, it is to lay the swaths side by side. Most of the guidance systems concentrate on keeping the tractor at a constant distance from the adjacent driving line, even if the objective is to consider the functional point of the implement instead of a tractor's position. In the case of a seed drill, the functional point is at the location of the coulters, in a sprayer the nozzles constitute the functional point, in a combine harvester the cutter head constitutes the functional point, and so forth. In a combined guidance system, the objective is to keep both the tractor and the implement on target line. Usually, some weighting parameters are used to set the importance of the targets. Generally speaking, a deviation of the implement from the target line is more important than a tractor deviation.

In the test configuration, two sensor sources for positioning were used. A *global*, GPS–based, RTK-type receiver was used in the tractor (Trimble 5700 VRS). The positioning and heading of the GPS receiver were improved by using inertial measurements (IMU, Xsens MT9-B) by means of pose estimation. The other sensor source was a *local* measurement; in the seed drill, a sensor measured the relative distance to the adjacent swath. The sensor was based on a SICK LMS221 laser scanner and a small plough that makes an identifiable furrow in the top soil next to the swath.

Thus, the test configuration had three actuators under control and two sensor systems. From a control engineering point of view, a multiple-input, multiple-output (MIMO) controller needed to be designed. The inherent noise sources for the system were wheel slip, the sensor noise, and the uncertainty of the model parameter (e.g. the varying total mass of the seed drill).

In the combined guidance system, the third part, in addition to the tractor and implement (from different manufacturers), is the *guidance system controller*. In the test configuration, the guidance system controller was based on a standard desktop computer (Intel DG45FC motherboard, Intel Core 2 Duo E8600 processor, 2 GB memory and Kvaser LeafLight HS CAN interface). The guidance system controller was connected to an ISO 11783 network and all of the commands and most of the measurements were transferred through that interface.

5. The structure of the navigation system

A navigation system is generally divided into four different layers of operations: *operation (or task) planning*, *path planning*, *path tracking* and *actuating*. In this case, the last layer was implemented in the tractor and could be controlled through the ISO 11783 network. In most cases, the first and perhaps the second layers are the responsibility of the human driver. But in the case of a *fully autonomous* tractor, these layers also have

Fig. 1 – The test configuration consisted of a standard tractor and towed trailer.

to be explicitly formulated and implemented. However, the operation planning for a farm management system still involves off-line planning, so it was excluded from the scope of this research.

5.1. *Components of the navigation system*

The physical structure of the navigation system consists of positioning devices, actuating devices and navigation devices. The positioning devices in this research included a ground-speed sensor (radar) for speed measurement, an inertial measurement unit (IMU) for roll and pitch-angle measurements, a real-time kinematic (RTK) GPS for accurate position measurements and a 2D laser scanner for local position measurements. The actuating devices included a steering controller, a cruise controller and a hydraulic valve for implement control. The navigation devices were the position estimation and guidance controllers. All of these devices were connected together through the ISO 11783 network. Figure 2 depicts the physical structure of the navigation system used in the test configuration.

The position estimation controller (labelled as *Navix*) was used mainly to estimate the yaw angle more precisely than the RTK GPS does. It used speed, position and angle measurements and an Extended Kalman Filter (EKF) to estimate the yaw angle. Also, the speed estimate was obtained from this filter. This type of position and heading estimation system has been discussed by Oksanen, Linja, and Visala (2005). The global position was measured using the RTK GPS. The RTK GPS positioning was then converted into the tractor coordinate system origin (the centre point of the rear axle at ground level) with the help of roll and pitch angles. The roll and pitch angles were collected directly from the commercial IMU.

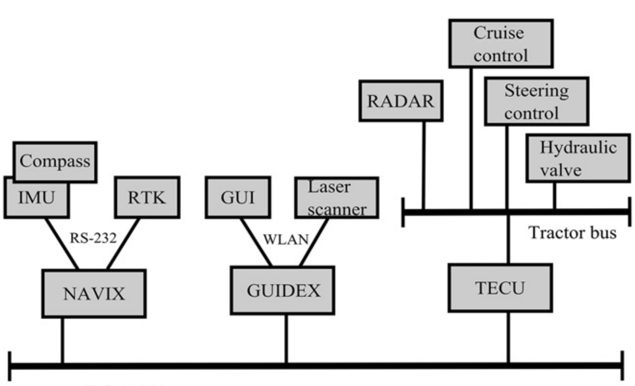

Fig. 2 – The physical architecture of the navigation system was built upon an ISO 11783 network.

The guidance system controller (labelled as *Guidex*) was the main element of the navigation system. It contained the software that calculates the control commands and sends them through the ISO 11783 network to the actuating devices (ISO 11783-7:2009). The Graphical User Interface (GUI) of the navigation system was connected to the Guidex through Wireless Local Area Network (WLAN). In the test configuration, Guidex was also used for handling the laser scanner measurements.

Logically, the physical components of the system are as follows: the tractor, the implement and the guidance system controller. The measuring devices of the tractor state and the components of the steering system were grouped into a single logical unit called simply the *Tractor*, whereas the measuring devices of the implement state and the components of the implement control were grouped into another logical unit called the *Implement*. This was done despite the fact that the mechanical control components were still located inside the tractor (the hydraulic valve). The logical grouping is depicted in Fig. 3.

Messages between these components were transferred through the ISO 11783 network. The messages sent between the Tractor and the Guidance computer were the position and the orientation messages, the speed information, the steering angle measurement and the control messages. These are all standard ISO 11783 messages (ISO 11783-7:2009).

In a headland operation, the implement has to be changed from a working state to an inactive state and back again after a turning manoeuvre. Since full autonomy is the objective, the navigation system should command the state of the implement automatically. In the test configuration, the state

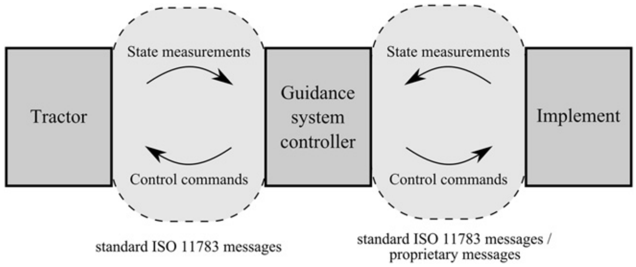

Fig. 3 – The logical architecture of the system from a navigational point of view.

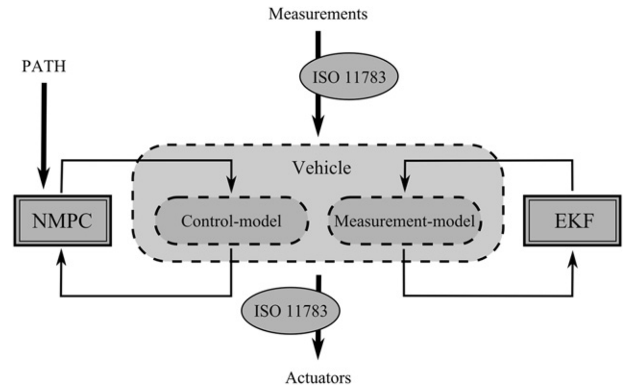

Fig. 4 – The logical structure of the software architecture has two concurrent loops: the estimation loop (EKF) and the optimisation loop (NMPC); the main information flows from the measurements to the actuators through the ISO 11783 network.

measurement and control were implemented by using proprietary messages (ISO 11783-1:2007; ISO 11783-3:2007).

In the test configuration, the implement also had a controllable joint in the drawbar. The low-level controller for this joint was implemented in the Guidance system controller and the control messages consisted of the hydraulic valve commands (ISO 11783-7:2007). The actual angle is needed both for the controller and for the state estimation. Currently, the ISO 11783 standard does not support these kinds of messages directly. That is the reason why the state information had to be carried out using proprietary messages. The standard, however, allows for proprietary messages (ISO 11783-1:2007; ISO 11783-3:2007).

The messages sent between the different components in the navigation system are listed in Table 1. The table also lists the physical device that is sending the message in question as well as the Parameter Group Number (PGN) number and the standard that defines the message.

5.2. *Software architecture*

The main goal was to get the software architecture for the navigation system to support all of the machinery configurations, while still remaining as simple as possible. The logical structure of the software is depicted in Fig. 4. The software has two concurrent loops: the *estimation* loop and the *optimisation* loop.

The software architecture consists of four different modules: *PATH*, *NMPC*, *EKF* and *Vehicle*. The EKF and Vehicle modules are involved in the estimation loop, whereas the PATH, NMPC and Vehicle modules are involved in the optimisation loop. The information carried between these modules and loops is transferred through the Vehicle module. For the experimental implementation developed here, all of the modules were implemented using the object-oriented C++ programming language.

5.2.1. *The NMPC module*

The Nonlinear Model Predictive Control (NMPC) was used to optimise the control variables. Many software packages are available for the NMPC. In this case, the Huge Quadratic Programming (HQP) software library (Franke & Arnold, 2008) was selected. It solves nonlinearly constrained problems with

Table 1 – Messages sent between different components in the navigation system.

Function/information	Device	From – To	PGN (hex)	Standard
Attitude (roll and pitch)	Navix	Tractor – Guidance	1F119	IEC 61162-3
Position	Navix	Tractor – Guidance	1F801	IEC 61162-3
COG&SOG (yaw and speed)	Navix	Tractor – Guidance	1F802	IEC 61162-3
GNSS Position data	Navix	Tractor – Guidance	1F805	IEC 61162-3
GNSS Pseudo noise statistics	Navix	Tractor – Guidance	1FA06	IEC 61162-3
Measured curvature	Steering control	Tractor – Guidance	AC00	ISO 11783-7
Setpoint curvature	Guidex	Guidance – Tractor	AD00	ISO 11783-7
Measured wheel speed	Cruise control	Tractor – Guidance	FE48	ISO 11783-7
Measured ground speed	Radar	Tractor – Guidance	FE49	ISO 11783-7
Response setpoint speed	Cruise control	Tractor – Guidance	FE0A	ISO 11783-7
Setpoint speed	Guidex	Guidance – Tractor	FE0B	ISO 11783-7
Measured work state	Implement	Implement – Guidance	FF16	*Proprietary*
Setpoint work state	Guidex	Guidance – Implement	FF15	*Proprietary*
Measured drawbar angles	Drawbar	Implement – Guidance	FF13	*Proprietary*
Measured lateral distance	Laser scanner	Implement – Guidance	FF14	*Proprietary*
Estimated valve flow	Valve	Tractor – Guidance	FE1x	ISO 11783-7
Setpoint valve flow	Guidex	Guidance – Tractor	FE3x	ISO 11783-7

a sequential quadratic programming (SQP) algorithm. Convex quadratic subproblems are solved using an interior-point method. The HQP is programmed using the C++ programming language. In addition, it uses the Meschach C library for sparse matrix computations. The original interface of the HQP was modified and a wrapper was made for it as a part of this research project. An interrupt routine was also added in order to ensure strict time limits. In the case of the interrupt, the old control values calculated in the previous time step were used and the optimisation horizon was reduced. The optimisation horizon is lengthened again in a step-by-step fashion when the HQP is able to solve ten consecutive optimisation problems without interruption. The scheduler is responsible for keeping the cycle time constant, and it generates the interruptions if the HQP does not end up with a feasible solution before the cycle time ends. The NMPC is described in more detail in Backman et al. (2012a).

5.2.2. *The EKF module*

The Extended Kalman Filter (EKF) was used for the state prediction because the state of the controlled system could not be directly measured. The obtained measurements were delayed for a specific amount of time. Also, the control outputs were delayed, and that affected the real system after some time. The NMPC controller needs an accurate estimate of the state when the current control outputs affect the controlled system. Otherwise, the stability of the controller cannot be guaranteed. Backman et al. (2012a) describe the state estimation in more detail.

5.2.3. *The PATH module*

The PATH module was responsible for the higher-level planning, for keeping track of the traversed swaths and for determining how the area of the field should be processed. The current desired working state of the system (stop, working, headland, transfer) can also be obtained from the PATH module. In the experimental navigation system, a simplified path planning algorithm was used. However, the implementation of the path planning algorithm also allows for more sophisticated planning algorithms.

First, the simplified path planning algorithm either takes the field outline from the data storage or the driver drives the first circuit around the field. Then, the previous driving line is followed at a constant distance. The process is repeated until a predefined number of circuits have been completed. After that, the longest edge of the field is identified and the previous driving line is followed until the longest edge has been reached. Finally, the rest of the field is processed by driving to and fro parallel to the longest edge, making turns either to the adjacent swath or by following a predefined turning pattern. The headland turnings are generated by a modified Dubins' Curves algorithm, which also ensures an upper-bounded curvature derivative. The path generation and planning algorithms are described in more detail in Backman, Oksanen, and Visala (2012b).

5.2.4. *The vehicle module*

The Vehicle module was actually a data storage and information centre. Other modules exchange information through this module. It also consisted of the information from the controlled system: the kinematic model and all the parameters.

The *Model* class is an implementation of a generic state space model. It is a storage class for the current state estimate, controls and measurements, but it also includes methods for state transition and calculating the Jacobian and state cost. The *Model* class also has a generic interface with both the NMPC and the EKF modules, so all of the models that are inherited from this class can be used for the estimation and the optimisation process. The NMPC and the EKF modules gain all the information they need from this class structure (Fig. 5).

Since the basic model of the system has been kept simple and exchangeable, there are also some auxiliary classes. These include *ModelConnect* for interconnecting two models, *ModelDelay* and *ModelIntegrator* for basic operations and *ModelMeasureDelay* for measurement delays. All of these auxiliary classes are also inherited from the *Model* class, so *ModelConnect* can be used to connect models and other auxiliary classes that have already been connected. In this way, there can be multiple simple kinematic models for the tractor and for the implement. The tractor or the implement can be changed and

Model
#_x: VECP #_u: VECP #_y: VECP #_f: VECP #_Q: MATP #_R: MATP #_dt: double
<<create>>-Model() <<destroy>>-Model() +dt(): double +dt(d: double): double +stateTransition(x: VECP, u: VECP, f: VECP): void +measure(x: VECP, y: VECP): void +step(): void +stateDifference(x: VECP, u: VECP, x_dif: VECP, x_der: VECP, u_dif: VECP, u_der: VECP): void +getWeights(Q: VECP, R: VECP): void +getStateJacobian(x: VECP, u: VECP, A: MATP): void +getControlJacobian(x: VECP, u: VECP, B: MATP): void +getMeasureJacobian(x: VECP, C: MATP): void +getStateNoise(Q: MATP): void +setStateNoise(Q: MATP): void +getInitialStateNoise(Q: MATP): void +getMeasureNoise(R: MATP): void +setMeasureNoise(R: MATP): void +getQualityVec(quality: VECP): void +getInitialMeasureNoise(R: MATP): void +getState(x: VECP): void +setState(x: VECP): void +getStateSize(): int +getInitialState(x: VECP): void +getStateLimits(xmin: VECP, xmax: VECP): void +getControl(u: VECP): void +setControl(u: VECP): void +getControlSize(): int +getControlLimits(umin: VECP, umax: VECP): void +getMeasure(y: VECP): void +setMeasure(y: VECP): void +getMeasureSize(): int +getMeasureDelay(d: VECP): void

Fig. 5 – The variables and methods of the *Model* class.

the rest of the model can be left untouched. Furthermore, the delay times of all of the measurements can be set independently.

Figure 6 shows the structure of the Vehicle module and how the information flows from and to the NMPC and the EKF modules. The *Tractor* and the *Implement* models are deepest inside the *Vehicle* class. These models are interconnected and together they comprise the *Kinematic* model. This model is further interconnected with the *MeasureDelay* and together they comprise the *Measurement* model. Separate from all that, the *Kinematic* model is also connected to the *Integrator* and together they comprise the *Control* model. The *Vehicle* class itself is not derived from the *Model* class.

5.2.5. *Scheduling*

Because the software was based on modules and also included several concurrent loops, some kind of scheduler was needed for to synchronise the loops. In order to keep the portability to different operation systems, the Boost C++ library (Boost, 2012) was used to create different threads and to handle barriers and locks between the threads.

During the start-up phase, the main thread is responsible for creating and initialising all of the necessary modules. After every module is up and running properly, they are divided into four different threads: the thread for handling the laser scanner measurements, the main thread, the NMPC thread and the path planning thread. The thread for handling laser scanner measurements is independent of all of the other threads. It runs when the new measurements arrive from the laser scanner and gives a new estimate of the swath position whenever the calculation is ready. The other threads are synchronised more precisely. The main thread is responsible for keeping the cycle time constant, in this case at 100 ms. The CAN-bus interface, the EKF calculation and the GUI messaging are in the main thread. The NMPC has its own thread for the optimisation step and the path planning layer also has its own thread for time-consuming tasks. If the NMPC calculation is not completed before the main thread reaches the synchronisation point, the NMPC calculation is interrupted. Otherwise, the NMPC waits for the new state estimate before the new calculation cycle begins. Figure 7 clarifies the timing schedule.

6. Results and discussion

The objective of this article was to discuss and present information on how a combined guidance system can be realised by using the ISO 11783 network, and furthermore, to discuss how the ISO 11783 network should be improved to better support a combined guidance system. The other objective was to propose a *decentralised* and *generic* guidance system for tractor–implement system.

As discussed above, the ISO 11783 standard contains a remote control message for commanding tractor steering by standard means. Furthermore, the standard makes it possible to obtain crucial information from the vehicle and from the GNSS system.

In the approach presented here, the measurements and the actuators were distributed and the messaging went through a common bus so decentralisation according to these components can be accomplished, which was

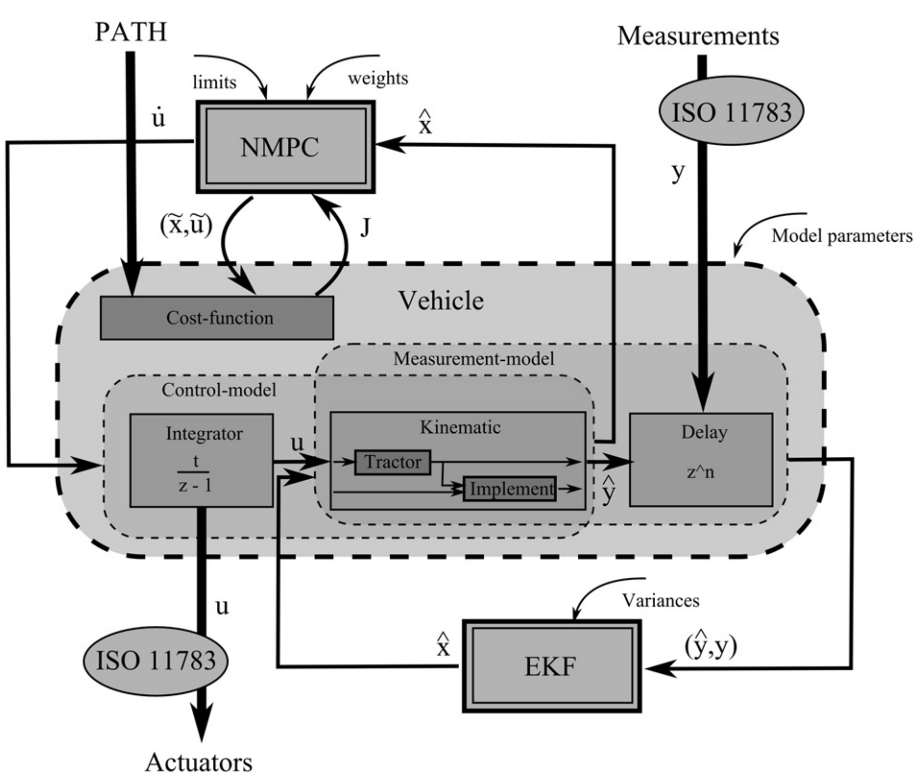

Fig. 6 – Information flows inside the software between the storage classes and the different modules (the vector contents are discussed in more detail in Backman et al., 2012a).

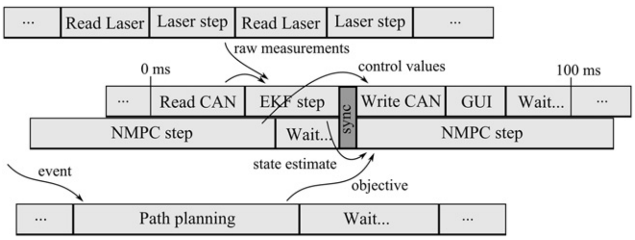

Fig. 7 – The scheduling diagram for all three threads: the thread handling the laser scanner measurements, the main thread, the NMPC thread and the path planning thread.

Table 2 – Some of the identified time-delay and time-constant parameters of the system and theoretical maximum latencies of the corresponding control and measurement messages.

Description	Identified		CAN latencies	
	Time-delay	Time-constant	Control	Measurement
Position	300 ms	–	–	1.0 ms
Heading	500 ms	–	–	1.0 ms
Speed	100 ms	740 ms	6.5 ms	16.5 ms
Steering	100 ms	120 ms	2.0 ms	1.5 ms
Free joint angle	200 ms	–	–	17.5 ms
Controlled joint angle	200 ms	450 ms	13.0 ms	17.5 ms

demonstrated via the test configuration. The guidance control system can only be decentralised in cases where the controllers of the tractor and the implement are separated, i.e. the tractor is steered based on a GNSS receiver placed on top of the tractor and the implement is steered based on another global or local measurement on the implement side. However, this trivial solution cannot be considered satisfactory for a true combined guidance system because it does not require an exchange of information. The presented NMPC algorithm is to realise a true MIMO controller for steering both the tractor and the implement in an optimal manner. For the NMPC, the information needs to be processed using a single processor because the state and model are needed from both the tractor and the implement. There are methods for decentralised the NMPC as well (Scattolini, 2009), but none of them are applicable for combined navigation because the information flow is limited and the trailer state influences the tractor control only through the global cost function and combined model. The state estimation could be distributed, but standardising state estimate transmission throughout an ISO 11783 network would be complicated. Therefore, the guidance system controller cannot be decentralised in a generic case.

The main result was that the navigation system could be built on top of the ISO 11783 network and that the architecture itself works. The accuracy results have been reported by Backman et al. (2010, 2012a). The parameters that affect the stability and that are partially caused by the ISO 11783 network are the delays. The time-delay and time-constant parameters of the system that were identified experimentally are listed in Table 2. The other system parameters are reported in more detail in Backman, Oksanen, and Visala (2011). Table 2 also shows the calculated maximum latencies of the corresponding messages in the ISO 11783 network that were used in the identification process. The theoretical maximum latencies are calculated according to equations found in Tindell, Burnds, and Wellings (1995). The impact of the CAN-bus latency is insignificant compared to the total dynamic delay. However, because the ISO 11783 network is an open system and a farmer can connect any number of machines to it, the maximum latencies cannot be guaranteed in every circumstance. The values in Table 2 are calculated for the test configuration where the CAN-bus load was approximately 30%.

Backman et al. (2012a) tested the accuracy of the navigation system by following straight driving lines and curved driving lines at different speeds. The mean lengths of the NMPC prediction horizon in the different tests are listed in Table 3. The prediction horizon was not reduced from the maximum only in the straight driving line test at a speed of 8 km h^{-1}. In all other tests, the computation time of the NMPC with a full-length prediction horizon was longer than the control cycle and the horizon had to occasionally be reduced. This implies that the NMPC controller cannot be used to realise real-time control in a tight loop without having any external interrupt or backup method to ensure strict time limits. Also, a permanent reduction of the prediction horizon could lead to decreased accuracy in navigation.

In the test configuration, the laser scanner measurement handling and the lower-level controller of the controllable joint were implemented in the guidance computer to save development resources. However, there is no reason why these cannot be distributed on the implement side as functions of the implement.

The ISO 11783 standard is not finished yet; several amendments and revisions are under preparation. A new part is also being developed (ISO/DIS 11783-14:2009) that will specify the messages and procedures for handling the headland automation. With the proposed part, the operator could record a sequence and store it to a specific computer called the Automated Functions Master. The sequence would be launched either automatically according to some trigger or by the user. This automation could be used to change the state of the implement in the headland instead of the proprietary messages. The combined guidance system should also be equipped with this standard part to better handle the headland operation of both the tractor and the implement.

Table 3 – Mean length of the prediction horizon under different conditions.

	Driving speed			
	8 km h^{-1}	10 km h^{-1}	12 km h^{-1}	14 km h^{-1}
Straight line	30	29.8	28.0	–
Curved line	28.2	27.6	26.0	24.6

7. Required changes to the ISO 11783 standard

The ISO 11783 standard makes generalisations about the different tractor kinematics in such a way that the quantity used for guidance 'steering' is curvature (the unit is km^{-1}). This generalisation supports front-wheel steering systems, rear-wheel steering systems and tracked vehicles as well as articulated tractors. However, for guidance purposes, it is important for the guidance controller to know the offset from the functional point to the rotating point of the vehicle.

The generic framework for a combined guidance system is a tractor and a single implement connected to the rear of the tractor. To be generic and support most of the common structures in modern machinery, up to four types of active steering systems on the implement side are considered: A) a hitch mounted with a side shift, B) a passive trailer and a drawhook side shift, C) a trailer with an articulated joint in the drawbar and D) a trailer with steering wheels. The four types are presented in Fig. 8.

For standardisation, a more generic quantity is needed; it should be simple enough to be generic, but at the same time give precise enough information for a combined guidance controller. The quantity of the 'steady state side shift', S_{ss}, is proposed. Figure 8 shows how the 'steady state side shift' should be considered for each type of controller. The steady state side shift should correspond to the side shift of the implement's functional point from the straight driving line that the tractor is following.

In the proposed concept, the combined guidance controller commands are implemented by sending a setpoint for the S_{ss}, and the implement has to use its proprietary means to realise the setpoint. Practically speaking, the implement may have its own hydraulic power system, which is operated by the tractor's Power Take-Off (PTO), or the implement may utilise ISO 11783 Tractor Class 3 remote control messages to control the hydraulic valves of the tractor, or by other means.

In the proposed framework, an implement has to transmit two measurements to the combined guidance system. The first is an estimated theoretical steady state side shift (S_{ss}), and the second is the measured angle of a free joint of the trailer. The estimated theoretical steady state side sift correspond to the setpoint, which means that the implement does not need to know the actual side shift, i.e. a straight driving line with slippage-free conditions can be assumed. It can be computed based on the position of the cylinder or other mechanical position measurements on the implement side.

In addition to communicating the setpoint measurements for real-time control, the message layout must also issue commands for the active/non-active quantities and any other similar quantities that are used in the ISO 11783 tractor guidance message. Furthermore, the NMPC and other similar controls require knowledge about the physical dimensions of the implement and its limits. For this purpose, the ISO 11783:10 already defines the offset and coordinate framework that can be used. However, more details about the physical limits of the implement are needed for a guidance system, e.g. the maximum steady state side shift. The mandatory fields required by a combined guidance system need to be defined. Information about the offsets and the other coordinate systems in the implement are transmitted to a combined guidance system by using the ISO 11783:10 means, in the device description.

8. Conclusions

A *combined guidance* system concept, where both a tractor and an implement were steered in the field to lay swaths side by side by means of the ISO 11783 communication standard, has been provided. A case study using a tractor and trailer-type

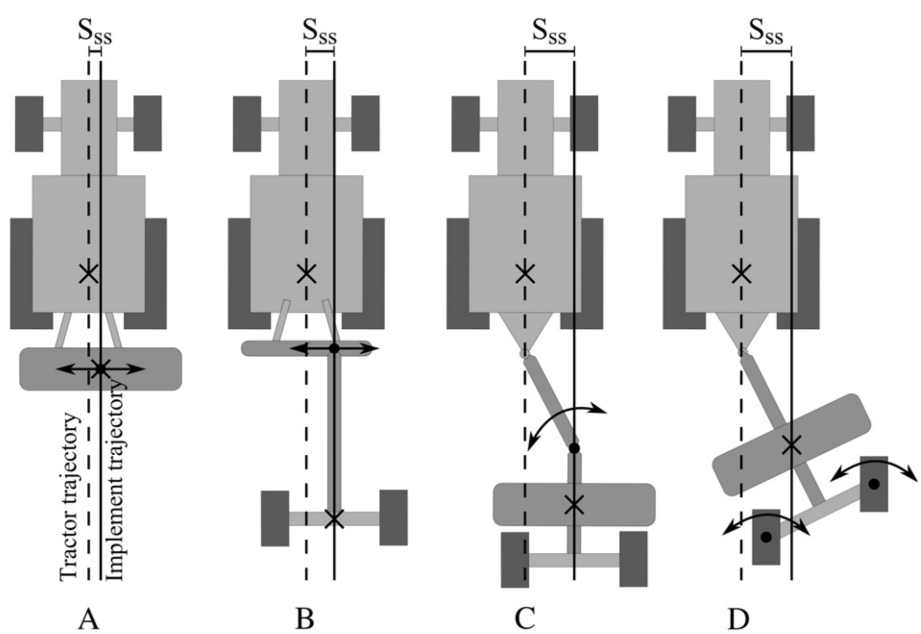

Fig. 8 – Kinematic types for implements with active steering: A) a hitch mounted with a side shift; B) a passive trailer and a drawhook side shift; C) a trailer with an articulated joint in the drawbar; D) a trailer with steering wheels.

seed drill indicates the benefits and the required technology for operation. The preliminary results show the benefits of a combined guidance system.

The article discussed the technology required to realise a combined guidance system. The underlying algorithm, a Nonlinear Model Predictive Control (NMPC) for handling multiple degrees of freedom, was proposed. This algorithm requires a kinematic model for the vehicle and also some dynamic parameters. The algorithm requires also a method to estimate the current state of the model as well as a method to calculate the control objective, i.e. a path to be tracked.

By using the case study, the information flows required for a combined guidance system with NMPC were illustrated. The information that can be transmitted between a tractor, an implement and a combined guidance controller using ISO 11783 standard messages is defined and what cannot be transmitted. The conclusion is that the information flows inside the combined guidance controller cannot be transferred easily over the ISO 11783 network, i.e. the combined guidance controller cannot be decentralised in the general case. However, all other information, including different measurements and controls, can be transferred over the ISO 11783 network.

As a result, the article proposes adding additional messages to the ISO 11783 standard that supports a combined guidance system as a manufacturer-independent solution. The ISO 11783 already supports remote control messages for the tractor. Similar messages for implement control are proposed. These messages include a concept of a unified quantity called 'steady state side shift', S_{ss}, for abstracting different kinematic types. Also additional information of the implement's physical dimensions and limits are required in the device description.

Acknowledgements

A part of this research was done in the Agromassi project, which is a part of FIMECC program EFFIMA. Part of this research was also funded by the Graduate School in Electronics, Telecommunications and Automation (GETA). GETA is a post-graduate programme offered jointly by five universities: the Aalto University School of Electrical Engineering, Tampere University of Technology and the Universities of Oulu, Turku and Jyväskylä.

REFERENCES

AGCO Advanced Technology Solutions. (2012). Topcon system 150 auto steering. Available online at http://www.agcotechnologies.com/en/autoguide.htm Accessed 25.04.12.

Arkin, R. C., & Balch, T. (1997). AuRA: principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence, 9*, 175–189.

Backman, J., Oksanen, T., & Visala, A. (2010). Nonlinear model predictive trajectory control in tractor–trailer system for parallel guidance in agricultural field operations. In *Proceedings of the agricontrol 2010*, Kyoto, Japan.

Backman, J., Oksanen, T., & Visala, A. (2011). Semi-automatic tuning of NMPC based trajectory control system in agricultural machine. In *Proceedings of the 18th IFAC world congress*, Milano, Italy (pp. 2912–2918).

Backman, J., Oksanen, T., & Visala, A. (2012a). Navigation system for agricultural machines: nonlinear model predictive path tracking. *Computers and Electronics in Agriculture, 82*, 32–43.

Backman, J., Oksanen, T., & Visala, A. (2012b). Path generation method with steering rate constraint. In *Proceedings of the 11th international conference on precision agriculture*, Indianapolis, Indiana, USA.

Blackmore, B. S., Griepentrog, H. W., Nielsen, H., Nørremark, M., & Resting-Jeppersen, J. (2004). Development of a deterministic autonomous tractor. In *CIGR international conference*, Beijing, China.

Blackmore, B. S., Have, H., & Fountas, S. (2002). A proposed system architecture to enable behavioural control of an autonomous tractor. In Q. Zhang (Ed.), *Automation technology for off-road equipment* (pp. 12–23). USA: ASAE.

Boost. (2012). Boost C++ libraries. Available online at http://www.boost.org Accessed 25.04.12.

CAN specification version 2.0. (1991). Robert Bosch GmbH.

Darpa. (2007). Darpa Grand Challenge. Available online at http://archive.darpa.mil/grandchallenge05/ Accessed 25.04.12.

Dickmanns, E. D. (2007). *Dynamic vision for perception and control of motion*. London: Springer.

Ehrl, M., & Auernhammer, H. (2007). X-by-wire via ISOBUS communication network. *Agricultural Engineering International: The CIGR Ejournal, 9*, 12.

Farm Journal Inc.. (2011). Today's auto-steering systems. http://www.agweb.com/assets/1/6/TodaysAuto-SteeringSystems_2011MG.pdf Accessed 06.11.12.

Franke, R., & Arnold, E. (2008). HQP: a solver for sparse nonlinear optimization. Available online at http://hqp.sourceforge.net/ Accessed 25.04.12.

IEC 61162-3. (2008). *Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 3: Serial data instrument network*. Geneva, standard.

ISO 11783-1. (2007). *Tractors and machinery for agriculture and forestry – Serial control and communications data network – Part 1: General standard for mobile data communication*. Geneva, standard.

ISO 11783-3. (2007). *Tractors and machinery for agriculture and forestry – Serial control and communications data network – Part 3: Data link layer*. Geneva, standard.

ISO 11783-7. (2009). *Tractors and machinery for agriculture and forestry – Serial control and communications data network – Part 7: Implement messages application layer*. Geneva, standard.

ISO 11783-9. (2002). *Tractors and machinery for agriculture and forestry – Serial control and communications data network – Part 9: Tractor ECU*. Geneva, standard.

ISO/DIS 11783-14. (2009). *Tractors and machinery for agriculture and forestry – Serial control and communications data network – Part 14: Sequence control*. Geneva, draft.

John Deere. (2012). GreenStar Precision Solutions. Available online at http://www.deere.com/wps/dcom/en_US/products/equipment/greenstar_precision_solutions/greenstar_precision_solutions.page Accessed 25.04.12.

Jørgensen, R. N., Sørensen, C. G., Pedersen, J. M., Havn, I., Olsen, H. J., & Søgaard, H. T. (2006). Hortibot: an accessory kit transforming a slope mower into a robotic tool carrier for high-tech plant nursing – part I. In *ASABE annual international meeting*, Portland, Oregon.

Lenain, R., Thuilot, B., Cariou, C., & Martinet, P. (2005). Model predictive control for vehicle guidance in presence of sliding: application to farm vehicles path tracking. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 885–890). Barcelona, Spain: ICRA.

Lenain, R., Thuilot, B., Cariou, C., & Martinet, P. (2006). Mobile robot control in presence of sliding: application to agricultural vehicle path tracking. In *45th IEEE conference on decision and control*, San Diego, CA, USA (pp. 6004–6009).

Maciejowski, J. M. (2002). *Predictive control: With constraints*. Harlow: Pearson Education.

Murphy, R. R. (2000). *Introduction to AI robotics*. Cambridge, MA, USA: MIT Press.
Nagasaka, Y. (2009). An autonomous rice transplanter guided by global positioning system and inertial measurement unit. *Journal of Field Robotics, 26*(6–7), 537–548.
Oksanen, T., Linja, M., & Visala, A. (2005). Low-cost positioning system for agricultural vehicles. In *Proceedings of the 2005 IEEE international symposium on computational intelligence in robotics and automation* (pp. 297–302). Espoo, Finland: CIRA.
Omead, A. (1990). *Integrated mobile robot control*. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.
Rintanen, K., Mäkelä, H., Koskinen, K., Puputti, J., Sampo, M., & Ojala, M. (1996). Development of an autonomous navigation system for an outdoor vehicle. *Control Engineering Practice, 4*(4), 499–505.
Ruckelshausen, A., Klose, R., Linz, A., Marquering, J., Thiel, M., & Tölke, S. (2006). Autonomous robots for weed control [Autonome Roboter zur Unkrautbekämpfung]. *Journal of Plant Diseases and Protection, 20*, 173–180.
SAE J1939. (1994). *Recommended practice for a serial communication network (class C)*. Society of Automotive Engineers, Standard.
Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control – a review. *Journal of Process Control, 19*, 723–731.
Strentz, A., Dima, C., Wellington, C., Herman, H., & Stager, D. (2002). A system for semi-autonomous tractor operations. *Autonomous Robots, 13*(1), 87–103.
Tindell, K., Burnds, A., & Wellings, A. J. (1995). Calculating controller area network (CAN) message response times. *Control Engineering Practice, 2*(8), 1163–1169.